View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Kent Academic Repository

Proceedings of the Operational Research Society Simulation Workshop 2016 (SW16)
A.Anagnostou, K. Hoad, M. Kunc eds.

TOWARDS SELF-ADAPTIVE DISCRETE EVENT SIMULATION (SADES)

Dr. Kathy Kotiadis

University of Kent
Canterbury, Kent CT2 7NZ
K.Kotiadis@kent.ac.uk

ABSTRACT

Systems that benefit from the ongoing use of simulation, often require considerable input by the modeller(s) to update and maintain the models. This paper proposes automating the evolution of the modelling process for discrete event simulation (DES) and therefore limiting the majority of the human modeller's input to the development of the model. This mode of practice could be named Self-Adaptive Discrete Event Simulation (SADES). The research is driven from ideas emerging from simulation model reuse, automations in the modelling process, real time simulation, dynamic data driven application systems, autonomic computing and self-adaptive software systems. This paper explores some of the areas that could inform the development of SADES and proposes a modified version of the MAPE-K feedback control loop as a potential process. The expected outcome from developing SADES would be a simulation environment that is self-managing and more responsive to the analytical needs of real systems.

Keywords: Reuse, Automation, Real-time, Self-Adaptation.

1 INTRODUCTION

The practice of simulation has been categorised into three modes (Robinson, 2002) simulation as software engineering; simulation as a process of organisational change; and simulation as facilitation. The first mode, supports models with a wide range of goals requiring a very accurate representation of the real world with models requiring years to develop through software engineering and requiring ongoing maintenance. This mode, unlike the other two modes, would aim for a long term use of a model. The second mode models typically have well-defined goals, aim for a reasonable representation and are developed using 'off the shelf' software. In this mode the maintenance costs of the model soon outweigh the benefits associated with its continued use, making them useful in the short to medium term. The third mode, involves using a model to support the involvement of a group of stakeholders in the modelling process (Tako and Kotiadis, 2015). It would almost certainly involve an 'off the shelf' package and the model would be 'shelved' following the intervention. An inspection of recent simulation literature (Taylor et al, 2013) reveals that another mode of practice is emerging that aims to support *ongoing rapid decision making* but does not necessarily require or aspire to the extensive software engineering development process required for mode 1. Indeed, research that fits with this mode of practice includes the development of generic models for reuse (e.g. Bartholet el al, 2005; Pidd, 2002), real time simulations (e.g. Tavakoli et al, 2008) and automation of various aspects of the modelling process (e.g. Huang et al 2011). However to a large extent these research areas are being developed independently from each other. This paper proposes bridging these areas in order to develop Self Adaptive Discrete Event Simulation (SADES) which could potentially be more effective at meeting the aim of this emerging mode of practice.

SADES would sense the need to run a simulation, from a stream of real time data, and then automate the evolution of the modelling process leading to a solution space. The aim of this mode would be the ongoing use of DES models in order to continually support decision making, ideally with very limited support by the modelling team.

The aim of this paper is to provide a vision of SADES and a focus on the research areas that could be bridged to develop SADES, such as the ones mentioned earlier and areas outside of simulation and OR, such as autonomic computing and software engineering for self-adaptive systems. The paper also considers how SADES might work and puts forward some preliminary ideas based on the MAPE-K adaptation control loop originally put forward by IBM (Kephart and Chess 2003). The impact of SADES would be to enable systems that benefit from simulation to be even more responsive to changes and enable ultimately the practice of simulation to become more sustainable.

The paper is organised into four further sections. The next section puts forward the vision of SADES, though an example, in order for the reader to appreciate how the various streams of research, subsequently described, can support the development of a SADES. Next a number of research areas are explored that could support the development of SADES. In the following section the MAPE-K loop is used to explain how SADES might work. The paper concludes with some proposed areas for further research.

2 ILLUSTRATING THE VISION OF SADES

In discrete event simulation (DES), a modeller will typically use a model of a system to explore various 'what if' scenarios. For example, a model simulating a UK hospital Accident and Emergency (A&E) department might be used to explore the number of doctors needed to meet the 'four hour' waiting target set by the Department of Health. Once an optimal or desired scenario is identified, it is implemented leading to changes in the actual system. Even in cases where a simulation model is used for understanding, so that no particular scenario is implemented as such, the system or the system's goals are likely to evolve over time. If a simulation model is to be reused in such instances it would require some updating to reflect the changes in system of interest or its goals. Given the costs of developing a model and its usefulness as a decision aid, it would seem sensible to reuse a model by updating and maintaining the model to support future decisions. However, that is often not the case because of the associated costs, such as, involving a modeller. One way of extending the lifecycle of DES models is to automate their evolution, so that the role of a modeller is largely restricted to the development stage of the simulation. For example, the application of a SADES to a UK A&E (Accident & Emergency or Emergency department in US) would enable a new simulation to be triggered whenever real-time data indicate that the 'four hour' target has been breached (Figure 1). This could be achieved through sensor technology attached to patients or devices within an A&E as well as connecting to databases and the internet. Sensors could also be attached to staff and other A&E resources. A smart management system, in the example described for simplicity as Monitor & Control, would automatically go through the modelling process so that a current model of the system is automatically generated, which is then automatically run to produce a range of feasible scenarios that fit with current goals. This would allow the stakeholders to respond quickly to a range of problematic situations without incurring further modelling costs.

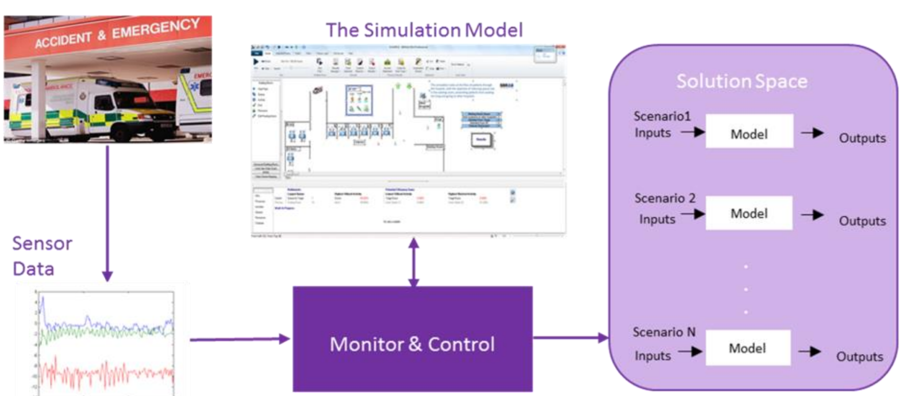

Figure 1 *An example of an A&E Self Adaptive DES model*

3 RELEVANT STUDIES

This section will explore some areas mentioned in the introduction that could initially drive the development of SADES. These areas are: DES reuse or building generic models, automation of DES processes, real time simulation and Self Adaptive Software Systems.

3.1 DES Reuse

Central to the philosophy of SADES is the idea of a long term use of a model or some aspect(s) of a model, which can be described as a form of reuse. Reuse is a concept that was discussed in the 60s in software (McIlroy, 1968) and as early as 1986 (Sargent, 1986) in simulation. Software reuse is the isolation, selection, maintenance and utilisation of existing software artefacts in the development of new systems (Reese and Wyatt, 1987; Robinson et al, 2004). Reuse in simulation spans a spectrum that ranges from full model reuse, to component reuse, to function reuse, to code scavenging (Pidd, 2002; Robinson et al, 2004). The benefits of reuse can be found in many forms such as saving effort, reducing development costs and enabling a quicker development time than developing a model from scratch (Waziruddin et al 2003).

More recently the idea of reuse has evolved to the creation of generic models that are built to fit a particular context. For example the District General Hospital Performance Simulator (DGHPSim) (Gunal and Pidd, 2011) is a (healthcare) domain specific example of reuse where a generic and restrictive model is built using data that would be found in a hospital's database such as a UK's Patient Administration Systems (PAS) and national databases such as the UK's Health Episode Statistics (HES). In another example, Pinto et al (2015) put forward a generic model for an ambulance system and even propose a generic method to develop such models.

Another aspect of reuse that has attracted interest is constructing complex systems from off-the-shelf components (Bartholet et al, 2005). Similar to whole model reuse, a simulation component that can be reused several times can save a great deal of time, money and human effort (Davis and Anderson, 2003). Indeed the benefits are even greater when reusing more than one component. Composability is about 'the capability to select and assemble simulation components in various combinations into valid simulation systems to satisfy specific user requirements' (Petty and Weisel, 2003). However reusing any component is not straight forward as it may only be valid under certain experimental frames e.g. a specific context (Zeigler et al, 2000). Spiegel et al (2005) explain that simulation composability and reuse will require comprehensive identification of constraints and where components are reused they must be designed with that in mind.

3.2 Automation in DES

A key aspect in the development of SADES is the automation of the evolution of the modelling process. The modelling process includes conceptual modelling, model coding, experimentation, implementation as well as validation and verification that takes places throughout the simulation lifecycle (Arthur and Nance, 2007). In principle there is reasonable agreement over what takes place during these stages but it is very hard to pin point these in detail for every study. Particularly for a process such as conceptual modelling, that is still considered by many to be an art. In this section we will explore some work on automation that addresses some of the activities that take place during the modelling lifecycle but for reasons of simplicity the real time simulation literature has been separated out, to be explored in the next section.

Data collection and analysis is largely an activity that relates to the initial stages in the modelling lifecycle. Padilla et al (2015) put forward a semi-automated initialization of input variables that are challenging to quantify and require additional processing to be assigned their initial values. The study supports the retrieval of data from structured and unstructured data sources and generates input data. They propose that further work could include exploring data from different sources such as data from social media and whether the assigned values yield more accurate simulation results (validation and verification). Schoogh et al (2010) also explore this area of DES automation and particularly so for manufacturing so that simulation models can be more frequently updated and integrated in the daily work of production engineers. Their study explores MTConnext tool for the collection of raw data and GDM-Tool for data processing (Bengtsson et al, 2009). They compare their approach to manual data handling and find that just for processing raw data they were able to reduce their work from half a day to 1 hour and conclude with future work to include some additional refinements for the tools.

Huang et al (2011) propose a data driven approach to automatically generate models using prebuilt and validated model components. In fact they view their approach as an automated reuse of model components. They provide an example of the automated generation of a rail network model using available data sources. They conclude that future research should focus on an automatic model calibration and different data analysis methods and techniques to estimate the models parameters.

A recent paper (Batarseh et al, 2015) contributes ideas on automation supporting model coding. The paper explores the use of model driven architecture (MDA) from software development to support the automatic creation of simulation models. The approach empowers domain stakeholders with tools for defining the problem to be solved and formally integrating the definition with the simulation tool. A key feature of the approach is expressing the semantics of the domain which is captured in a library. The authors suggest that further work on addressing warm up issues and transients is needed so that domain experts do not misuse the simulation tool. `

The stage of experimentation involves model calibration as well as searching the solution space. Hoad et al (2010a,b) put forward an automatable algorithm to select the number of replications and automated the process of estimating the warm up length for steady state systems which have since been incorporated within the SIMUL8 simulation package (www.simul8.com). Hoad et al (2011) put forward the AutoSimOA framework explored automating the analysis of simulation output. It includes generating replications, warm up and run length estimation.

Waziruddin et al (2003) put forward a study that could inform both automated model coding and automated experimentation (searching the solution space). Specifically they explored the process of 'coercing simulations' that involves a combination of code modification and simulation behaviour optimisation with the goal of driving the behaviour of a simulation so that it satisfies a new set of requirements. They propose that some aspects can be automated so that ultimately the transformation of a simulation to meet new requirements can be performed more efficiently.

3.3 Real time simulation and dynamic data driven simulations

One characteristic that SADES simulations should have is the ability to sense the need to run a new simulation. This would require the ability to automatically monitor real time data and process that data so that it can be used in the simulation. Hence it involves automation of processes that fit within conceptual modelling and model coding. Two areas have been identified as relevant in the development of these capabilities: real time DES and dynamic data driven simulations.

Real time DES are essentially simulation models using real or near to real time data rather than historical data and are meant to overcome shortcomings of traditional simulation (other modes). The shortcomings non real time DES (Tavakoli et al, 2008) are: a. being time consuming because of the manual collection and analysis of data, b. time depended on historical data that soon enough become obsolete, c. inaccurate for prediction because of issues with historical data and d. costly because of effort and expertise required for the data collection. Real time simulation modelling has been predominantly been explored within a number of contexts such as manufacturing and healthcare (Tavakoli et al 2008; Mousavi et al 2011), construction (Song and Eldin, 2012; Vahdatikhaki et al, 2013; Vahdatikhaki and Hammad, 2014) and road and traffic management (Henclewood et al, 2008; Henclewood et al, 2010; Henclewood et al, 2012; Jaoua et al, 2012a,b).

Most of the studies mentioned explain the functionality of their real time simulation using a framework or architecture. We will now explore some of these:

- An early study by Lee and Fishwick (1999) put forward a multimodeling methodology called OOPM/RT (Object Oriented Physical Modeler for Real Time Simulation) that minimises the modellers interventions, through a semi-automatic method, and hence supports the modelling of real time systems. The approach considers the question of how to determine the optimal model that simulates the system by a given deadline while still producing valid outputs at an acceptable level of detail. They identify some issues that were unresolved in OOPM/RT such as optimality (of the level of abstraction) and validation challenges.
- Tavakoli et al (2008) propose an architecture for data acquisition and data restructuring called Flexible Data Input Architecture (FDILA) which they illustrate with examples in manufacturing and healthcare. In a later study Mousavi et al (2011) put forward the SIMMON architecture that allows extensive data acquisition technologies that prepares the information for the real time simulation package (restructuring) and it subsequently translates the simulation results into easy to comprehend business performance parameters. The extension to this work mainly related to further testing. Particularly insightful for real time simulation is their work on input variable selection methods using a case study in the deep drilling industry (Tavakoli et al 2013).
- Vahdatikhaki and Hammad (2014) put forward a framework for near real time simulations of earthmoving projects that use location tracking technologies. The framework among other things captures different levels of sensory data, then processes it, analyses it and filters it. The authors suggest that future work will include enabling distribution fitting from the data captured and updating the functionality of the framework to include the possibility of updating the logic of the model and the sequence of activities when required. Similarly, Song and Eldin (2012) propose an adaptive real time tracking and simulation framework for heavy construction applications with look ahead scheduling. The idea here is to react to changes in real time or near to real time. They conclude with future work to include to finding a way to collect and transmit data only when a meaningful change occurs in the real system and to enable an automated generation of construction simulation models from scratch using extracted knowledge about key model parameters and precedence logic.
- Henclewood et al (2010) use real time sensor data e.g. from loop detectors or video cameras, to better manage traffic operations. In subsequent research, Henclewood et al (2012) conclude that further work is needed to develop new calibration procedures as models for traffic simulations are often calibrated using data from a single time period.

We now explore some ideas from dynamic data driven application systems (DDDAS) (Darema, 2004; Darema, 2005; Gaynor et al, 2005; Farhat et al, 2006; Kennedy and Theodoropoulos, 2006) given their potential to extend real time DES simulations and ultimately inform SADES. '*The vision of DDDAS, goes beyond the current concepts of real time control, in terms of the concept of interaction of the application simulation with the measurement system, the range of applications, the streamed input data and the scope of systemic approach to addressing the ensuing challenges: assessment of data quality, incorporation of uncertainty, ability to combine different kinds of data taken at different times, application algorithms with stable response to the streamed data, distributed systems (for sensing, for computation and for output, and for data storage), development of effective interfaces of applications with measurement processes, supporting the execution of such applications with dynamic resources requirements*' (Darema, 2004, 663-4).

DDDAS entail the ability to incorporate additional data (archived or real time) and for applications to dynamically steer the measurement process (Darema, 2004). DDDAS has been explored within a number of contexts, not always connected to a simulation, and is meant to be particularly useful in supporting crisis situations where rapid decision making is essential (Gaynor et al, 2005; Farhat et al, 2006, Yan et al, 2009). We could say that a simulation model supports situations when a problematic situation emerges (i.e. crisis) so a symbiosis is perfectly reasonable and has been explored within other types of simulation such as agent based simulation (Kennedy and Theodoropoulos, 2006). Indeed Henclewood et al (2008) propose a methodology for a real time (dynamic) data driven simulation. A key part of the research is the development of dynamic data driven application (agent based) simulation (Fujimoto et al, 2006).

3.4 Self-Adaptive Systems

In order for DES to self-adapt it needs to be able to automatically go through the modelling process and to automatically sense the system and its wider environment and generate a model of the same system, albeit in its new (possibly changed) state. This idea of self-adaptation for DES is taken from the field of self-adaptive (software) systems. Self-adaptive systems have been explored within the different areas of software engineering, as well as other related research communities, such as control engineering and autonomic computing (Brun et al, 2009). These communities use common language and ideas and the *self* prefix indicating the autonomous nature of the system. ‘The essence of autonomic computing systems is self-management, the intent of which is to free system administrators from the details of system operation and maintenance and to provide users with a machine that runs at peak performance 24/7.’ (Kephart and Chess 2003, p42). An analogy to this is the autonomic nervous system in the human body, which among other things enables the pupils to adjust to sunlight, enables sweating to keep cool, adjusts breathing rate etc. with no conscious recognition or effort from our part for all this to take place. Autonomic computing grew out of a need to prepare for the next generation of computing systems with increased interconnectivity and integration. Some of the original characteristics of autonomic computing include self-management, self-configuration, self-optimisation, self-healing and self-protection. Brun et al (2009) and more recently Macias-Escriva et al (2013) put forward a number of challenges with a particular emphasis on the design of a self-adaptive system and the importance of feedback loops.

Feedback loops provide the generic mechanism for self-adaptation and typically contains four key activities: *collect, analyse, decide and act* (Brun et al, 2009). The data typically *collected* from environmental sensors and other sources reflecting the current state of the system is cleaned, filtered, pruned and stored for future reference to reflect past and current states. The system subsequently *analyses* the data and then a *decision* is made about how to adapt the system so it reaches a desirable state. The final activity in the loop is to act by implementing the decision. The properties of the control loop will ultimately affect the system design, architecture and capabilities (Brun et al, 2009). Therefore it is important that the control loop and its properties are adequately explained.

One of the key feedback loops in software engineering is the MAPE-K which stands for Monitor, Analyse, Plan and Execute over a Knowledge base (Brun et al, 2009) as shown in Figure 2. In the feedback loop the autonomic manager and the managed element correspond to the controller and the process respectively. A key addition to the generic mechanism is the knowledgebase which is shared by the MAPE engine. The *monitor* senses the managed element, filters and stores the event data in the knowledgebase for future reference. The *analyser* compares event data against patterns found in the knowledgebase to diagnose and store symptoms for future reference in the knowledgebase. The symptoms are interpreted by the *planner* that subsequently devises a plan to *execute* a change in the managed element through its effectors. The autonomic manager, with the feedback loop at its core, will adjust the managed element if necessary according to its control objective (Brun et al 2009). In the subsequent section we will explore how this MAPE-K adaptation loop can be used to support the design of SADES.

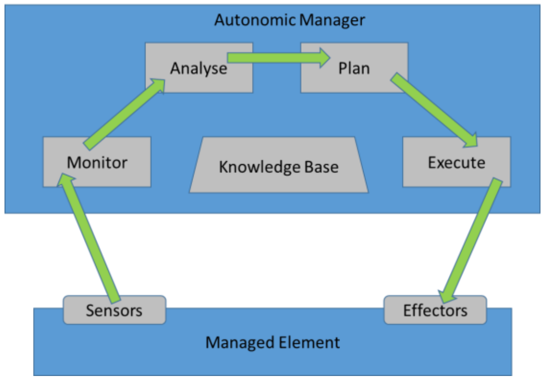

Figure 2 *The MAPE-K loop reproduced from IBM (Keller, 2005)*

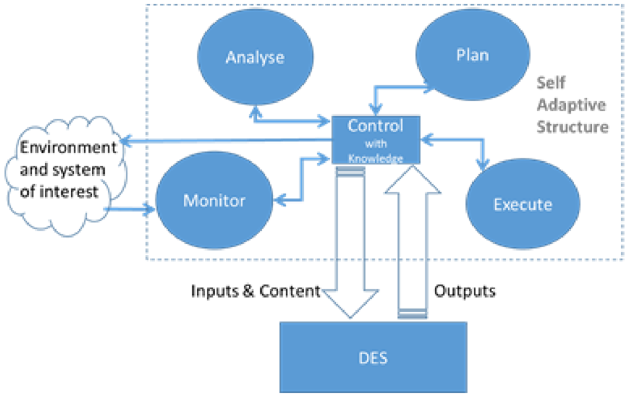

Figure 3 *Re-interpreting the MAPE-K structure for SADES*

4 A FEEDBACK STRUCTURE FOR SADES

This section will explore how the MAPE-K loop might be re-interpreted as a structure for SADES, on occasion referring to the A&E example. In the vision of SADES the evolution of the modelling process is automated. One view of how MAPE-K could be adapted for SADES can be seen in Figure 3 with the managed element being the simulation model. The SADES in Figure 3 differs mainly with respect to the loop in MAPE-K (Figure 2) because the simulation modelling process is not a linear activity. Activities relating to analysis for example, take place before the model is determined (e.g. analysis of raw data) as well as on completion of a model run (e.g. analysis of warm up time). Hence the interpretation of the MAPE-K loop for SADES takes the form of a centrally controlled structure (autonomic manager) rather than a loop.

The control of the sequence of the elements in the structure and the simulation model, assigned to the autonomic manager, is based on the information held in the knowledgebase (Control with Knowledge in Figure 3). The knowledgebase would be a storage for the data and information that is important to the evolution of the modelling process, such as the simulation objectives. The content of the knowledgebase could be updated either by the elements in the SADES structure, the autonomic manager or by human intervention. For example, the expectation for the objectives is that these could be automatically or manually altered over time to reflect changes to the needs of the system of interests e.g. the A&E department. We will now explore how the some of the activities in the DES modelling process could potentially be allocated to the four elements of the SADES control structure.

The Monitoring element would undertake the activities of sensing information about the environment and system of interest to be simulated. This corresponds to the data collection in the DES modelling process. This data would be filtered and stored in the knowledgebase for future reference. For example for a SADES of the A&E department, it might sense and collect data about the available human resources for each shift, the arrival, queuing and throughput times for each patient, the processes involved and information about the categorisation of need/severity of the emergency (e.g. low, medium, high). The data would need to be sorted to match up to shifts, days etc, possibly translated into a different time unit or format and stored in the knowledgebase for future reference. Data to be used for Validation and Verification could also be sensed and stored separately for that purpose in the knowledgebase. Other data sensed might be sourced from the web, for example relating to targets (e.g. 4 hour target) or epidemiological data relating to the target population of that A&E. The autonomic manager could also be designed to have capability to request new data to be sensed by this element if the objectives are altered and therefore alter what is monitored.

The *Analyse* element could undertake analytical activities on the data collected and held in the knowledgebase. For example, the analysis might involve calculating a moving average for the queuing time across each shift and compared to data held in the knowledgebase on the acceptable levels of the relevant KPI. KPIs would be matched up to the objectives of the A&E, held in the knowledgebase. The Analyser in the loop would also determine parameters for distributions or the distributions themselves for the simulation model. Additionally it could analyse the data for the warm up time and run length time (if necessary) and the number of replications. It would also analyse the output data from the resulting scenarios and statistically compare the scenarios. This element could also perform some comparisons of the analysed V&V data to the model output. The analysed data, distribution selection and parameters, calibration parameters, V&V comparisons, would be stored for future reference in the knowledgebase for planning and execution. Hence this element of the loop replaces DES analytical activities that are typically found in all the different parts of the modelling process.

The main roles of the planner or *planning* element could be that of: a. interpretation of the analysis, by consulting the knowledgebase, b. to determine if a simulation should be undertaken and c. to schedule subsequent modelling activities. For example, any breaches of the system (e.g. four hour target in A&E) KPIs would be interpreted by the *planner*, that would devise a plan to execute in the managed element (the simulation model). The *planner* would determine the model content, the experimental design by setting out the inputs and variation. The plan to be executed could include changes to: inputs (e.g. number resources, distributions & parameters), model content (e.g. type of resources, processes, rules etc.) which would be deposited in the knowledgebase. Hence this element of the structure replaces the DES modeller decision making activities that are typically found in all the different parts of the modelling process.

The execution element or *executor* could consult the plan held in the knowledgebase and translate the plan into sequential or concurrent actions to generate the model content, inputs and assign the calculation of KPIs for each scenario and ensure that the planned activities are executed in the correct order. These activities could also detail the interaction of the simulation output with the elements of the SADES structure. The executor instructions would be deposited in the knowledgebase for action.

The autonomic manager would initialise the simulation by following the execution instructions. The data from the simulation runs would be stored in the knowledgebase for subsequent analysis. It is envisaged that the autonomic manager might revisit elements of the SADES structure (possibly forming a loop) several times before the process is completed. At the end of the process, the solution space would have been captured in the knowledgebase. This in turn would be reported to the stakeholders by the autonomic manager. Ideally some data from the V&V comparisons would also be provided alongside the solution space.

5 CONCLUSION

Brun et al (2009) when referring to the studies undertaken in the many communities exploring self-adaptation describes these as isolated and from their own perspective. This is also true for the studies undertaken in DES aiming for ongoing and rapid decision making, but this study has set out to change this and provide a common focus in developing SADES. SADES could potentially follow the automation-based paradigm for DES which aims to reduce the cost and duration of software development; achieve maintainability; develop reusable software; increase user involvement and reduce software portability issues (Balci and Nance, 1987). Balci and Nance (1987) when proposing the paradigm suggested that it would require further technological developments. Advancements in software engineering and in the state of the art of modelling and simulation over the last two decades, lead us to predict that this mode is feasible providing the simulation community come together to bridge the existing research.

Indeed, studies on DES reuse or building generic models, automation of DES processes, real time simulation and Self Adaptive Software Systems could feed into the development of SADES. For example the studies in real time simulation reveal that a great deal has been accomplished with regard to collecting and preparing data for simulations that could feed into the monitoring element of the SADES (Figure 3). However some existing advancements are context specific and there are still other hurdles such as the automatic generation of models, sensing meaningful change in the data, calibration of models for different time periods etc. Similarly, a lot of the research from reuse could feed into SADES because it is entirely possible that previously generated models of that system could be reused. For example it could be reuse of code or components or some data. An aspect of reuse that will support the development of SADES is understanding how to develop or design components or models that will be reused in a modified state. Another aspect that will be very important in the development of SADES is composability, which is about combining and recombining components to meet different sets of requirements without substantial integration efforts (Bartholet et al, 2005). To sum up there are a considerable number of studies that could be used to inform the development SADES but this will be easier to achieve as part of a more collaborative research agenda. The envisaged impact of SADES on future systems that will be more connected (think of Internet of Things), is to offer a more sustainable and responsive mode of simulation. The expectation is that by bridging the relevant research we can better understand SADES, design it and ideally construct examples of it.

ACKNOWLEDGMENTS

Kathy Kotiadis acknowledges the support and the sponsorship of the Daphne Jackson Trust, the University of Kent and Simul8 in undertaking this research.

REFERENCES

Arthur JD and Nance RE (2007). Investigating the use of software requirements engineering techniques in simulation modelling, *Journal of Simulation*, **1**, pp 159-174.

Balci O and Nance RE (1987). Simulation Support: Prototyping the Automation-based Paradigm, *Proceedings of the 19th Winter Simulation Conference*, New York, NY, USA, pp. 495-502.

Bartholet RG, Brogan DC and Reynolds PF Jr. (2005). The computational complexity of component selection in simulation reuse, *Proceedings of the 37th Winter Simulation Conference*, Orlando FL. pp.2472-2481

Batarseh OG, Huang E and McGinnis L (2015). Capturing simulation tool and application domain knowledge for automating simulation model creation, *Journal of Simulation*, vol. 9, no. 1, pp. 1-15.

Bengtsson N, Shao G, Johansson B, et al (2009). Input Data Management Methodology for Discrete Event Simulation, *Proceedings of the 09 Winter Simulation Conference*, Austin, Tx, pp. 1335-1344

Brun Y, Marzo Serugendo G, Gacek C, et al (2009). *Engineering Self Adaptive Systems through feedback loops*, In: Software Engineering for Self-Adaptive Systems, eds. B.H. Cheng, R. Lemos, H. Giese, P. Inverardi & J. Magee, Springer-Verlag, Berlin, Heidelberg, pp. 48-70.

Darema F (2005). *Dynamic data driven applications systems: New capabilities for application simulations and measurements*, In Computational Science ICCS 2005 Springer, pp. 610-615.

Darema F (2004). *Dynamic data driven applications systems: A new paradigm for application simulations and measurements,* In Computational Science ICCS 2004 Springer , pp. 662-669.

Davis PK and Anderson RH (2004). Improving the composability of DoD Models and Simulations, *JDMS: Journal of Defense Modelling and Simulation*, 1(1), pp. 5-17.

Farhat C, Michopoulos J, Chang F, et al (2006). *Towards a dynamic data driven system for structural and material health monitoring*, In Computational Science ICCS 2006 Springer, pp. 456-464.

Fujimoto R, Guensler R, Hunter M, et al (2006). *Dynamic data driven application simulation of surface transportation systems*, In Computational Science ICCS 2006 Springer, pp. 425-432.

Gunal MM, and Pidd M, (2011). DGHPSIM: Generic Simulation of Hospital Performance, *ACM Trans Model Comput Simul*, 21 (4), pp. 23:1-23:22.

Gaynor M, Seltzer M, Moulton S, et al (2005). *A dynamic, data-driven, decision support system for emergency medical services*, In Computational Science ICCS 2005 Springer, pp. 703-711.

Henclewood D, Guin A, Guensler R, et al (2010). Real-time Data Driven Arterial Simulation for Performance Measures Estimation, *Proceedings of the 2010 Winter Simulation Conference*, pp. 2057-2069

Henclewood D, Hunter M, and Fujimoto R, (2008). Proposed Methodology for a Data-driven Simulation for Estimating Performance Measures Along Signalized Arterials in Real-time, *Proceedings of the 40th Conference on Winter Simulation Conference*, pp. 2761-2768.

Henclewood D, Suh W, Rodgers M et al (2012). A Case for Real-time Calibration of Data-driven Microscopic Traffic Simulation Tools, *Proceedings of the Winter Simulation Conference*, pp. 148:1.

Hoad K, Robinson S, and Davies R, (2011). AutoSimOA: A framework for automated analysis of simulation output, *Journal of Simulation*, 5 (1), pp. 9-24.

Hoad K, Robinson S, and Davies R, (2010a). Automated selection of the number of replications for a discrete-event simulation, *Journal of the Operational Research Society*, 61 (11), pp. 1632-1644.

Hoad K, Robinson S, and Davies R, (2010b). Automating warm-up length estimation, *Journal of the Operational Research Society*, 61 (9), pp. 1389-1403.

Huang Y, Seck MD, and Verbraeck A, (2011). From Data to Simulation Models: Component-based Model Generation with a Data-driven Approach, *Proceedings of the Winter Simulation Conference*, pp. 3724-3734.

Jaoua A, Gamache M and Riopel D, (2012). Specification of an intelligent simulation-based real time control architecture: Application to truck control system, *Computers in Industry*, 63(9), pp. 882-894.

Jaoua A, Riopel D, and Gamache M (2012). A simulation framework for real-time fleet management in internal transport systems, Simulation Modelling Practice and Theory, 21(1), pp. 78-90.

Keller A, (2005). Towards Autonomic Networking Middleware, Slides from Keynote talk at International Workshop on Next Generation Networking Middleware.

Kennedy C, and Theodoropoulos G, (2006). *Intelligent management of data driven simulations to support model building in the social sciences*, In Computational Science ICCS 2006 Springer, pp. 562-569.

Kephart JO, and Chess DM, (2003). The Vision of Autonomic Computing, *Computer*, 36(1), pp. 41-50.

Lee K, and Fishwick PA, (1999). OOPM/RT: A Multimodeling Methodology for Real-time Simulation, *ACM Trans ModelComput Simul*, 9 (2), pp. 141-170.

Macias-Escriva FD, Haber R, del Toro R et al (2013). Self-adaptive systems: A survey of current approaches, research challenges and applications, *Expert Systems with Applications*, 40(18), pp. 7267-7279.

McIlroy M (1968). Mass produced software components, Software engineering, *Report on a conference sponsored by the NATO Science Committee*, ed. P. Naur and B. Randell, Scientific Affairs Division, NATO, Brussels, October, pp. 138-155.

Mousavi A, Komashie A, and Tavakoli S, (2011). Simulation-based Real-time Performance Monitoring (SIMMON): A Platform for Manufacturing and Healthcare Systems, *Proceedings of the Winter Simulation Conference*, pp. 600.

Padilla JJ, Diallo SY, Kavak, H. et al (2015). Semi-automated initialization of simulations: an application to healthcare, *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*. first published on January 8, 2015 doi:10.1177/1548512914565503

Petty MD, and Weisel EW (2003). A composability lexicon, *Proceedings of the Fall 2003 Simulation Interoperability Workshop*, pp. 181-187.

Pidd M, (2002). Reusing Simulation Components: Simulation Software and Model Reuse: A Polemic, *Proceedings of the 34th Conference on Winter Simulation*, pp. 772-775.

Pinto LR, Silva PMS and Young TP (2015). A generic method to develop simulation models for ambulance systems, *Simulation Modelling Practice and Theory*, 51(0), pp. 170-183.

Reese R, and Wyatt DL (1987). Software Reuse and Simulation, *Proceedings of the 19th Conference on Winter Simulation*, New York, NY, USA, pp. 185-192.

Robinson S (2002). Modes of simulation practice: approaches to business and military simulation, *Simulation Modelling Practice and Theory*, 10(8), pp. 513-523.

Robinson S, Nance RE, Paul RJ, et al (2004). Simulation model reuse: definitions, benefits and obstacles, *Simulation Modelling Practice and Theory*, 12(7–8), pp. 479-494.

Sargent RG (1986). Joining Existing Simulation Programs, *Proceedings of the 18th Conference on Winter Simulation*, New York, NY, USA, pp. 512-516.

Skoogh A, Michaloski J and Bengtsson N, (2010). Towards Continuously Updated Simulation Models: Combining Automated Raw Data Collection and Automated Data Processing, *Proceedings of the Winter Simulation Conference*, pp. 1678-1689.

Song L, and Eldin NN (2012). Adaptive real-time tracking and simulation of heavy construction operations for look-ahead scheduling, *Automation in Construction*, 27(0), pp. 32-39.

Spiegel M, Reynolds J, Paul F et al (2005). A Case Study of Model Context for Simulation Composability and Reusability, *Proceedings of the 37th Conference on Winter Simulation conference*, pp. 437-444.

Tako AA and Kotiadis K (2015). PartiSim: A multi-methodology framework to support facilitated simulation modelling in healthcare, *European Journal of Operational Research*, 244(2), pp. 555-564.

Tavakoli S, Mousavi A, and Poslad S, (2013). Input Variable Selection in Time-critical Knowledge Integration Applications: A Review, Analysis, and Recommendation Paper, *Adv Eng Inform*, 27(4), pp. 519-536.

Tavakoli S, Mousavi A, and Komashie A, (2008). A Generic Framework for Real-time Discrete Event Simulation (DES) Modelling, *Proceedings of the 40th Winter Simulation Conference*, pp. 1931-1938.

Taylor SJE, Balci O, Cai W, et al (2013). Grand Challenges in Modeling and Simulation: Expanding Our Horizons", *Proceedings of the 2013 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, ACM, New York, NY, USA, pp. 403-408.

Vahdatikhaki F and Hammad A (2014). Framework for near real-time simulation of earthmoving projects using location tracking technologies, *Automation in Construction*, 42(0), pp. 50-67.

Vahdatikhaki F, Setayeshgar S, and Hammad A (2013). Location-aware Real-time Simulation Framework for Earthmoving Projects Using Automated Machine Guidance, *Proceedings of the 2013 Winter Simulation Conference*, IEEE Press, Piscataway, NJ, USA, pp. 3086-3097.

Waziruddin S, Brogan D, and Reynolds P, (2003). The Process of Coercing Simulations, *Proceedings of the Fall 2003 Simulation Interoperability workshop*.

Yan X, Gu F, Hu X, et al (2009). A Dynamic Data Driven Application System for Wildfire Spread Simulation, *Proceeding of the Winter Simulation Conference*, pp. 3121-3128.

Zeigler BP, Kim TG and Praehofer H (2000). *Theory of Modeling and Simulation*, 2nd edn, Academic Press, Inc, Orlando, FL, USA.

AUTHOR BIOGRAPHIES

KATHY KOTIADIS is a Daphne Jackson fellow at the University of Kent. She is a former assistant professor at the University of Warwick and has a PhD from the University of Kent. Her research interests include the process of modelling, conceptual modelling, discrete-event simulation, Soft Systems Methodology (SSM), facilitated approaches in healthcare and developing Self Adaptive Discrete-Event Simulation. Her publications can be found on https://www.researchgate.net/profile/Kathy_Kotiadis